Amendments to the Specification:

The specification has been amended to correct various typographical/grammatical errors.

Please replace paragraphs [0027] - [0029] with the following amended paragraphs:

[0027] Please continue to refer to Fig.4. In the present embodiment, the microprocessor 30 does not directly acquire the desired digital data from the serial flash memory 32 but utilizes the buffering/controlling device 38 beforehand to consecutively access a plurality of digital data in the serial flash memory 32 instead. The buffering/controlling device 38 outputs an operating clock to the microprocessor 30 so as to slow down/stop microprocessor 30; that is, when the operating clock disappears, the operations of the microprocessor 30 will suspend, when the operating clock [[slow]] slows down, the operations of the microprocessor 30 will also slow down. The buffering/controlling device 38 can consecutively access a predetermined number of programming codes in the serial flash memory 32. When the microprocessor 30 requires the programming codes and accesses the buffering/controlling device 38, the microprocessor 30 emits an access address corresponding to the programming codes to the buffering/controlling device 38 so that the buffering/controlling device 38 can judge whether the access address emitted from the microprocessor 30 is located in the buffering/controlling device 38. If the buffering/controlling device 38 stores the programming codes required by the microprocessor 30, the microprocessor 30 directly receives the desired programming codes from the buffering/controlling device 38. On the other hand, if the buffering/controlling device 38 does not store the programming codes required by the microprocessor 30 （for instance, the microprocessor 30 is in jump condition）, the buffering/controlling device 38 will stop the outputting of the operating clock so that the microprocessor 30 will suspend and retain the current conditions due to the

disappearance of the operating clock. In the meantime, the buffering/controlling device 38 transmits the access address corresponding to the programming codes to the serial flash memory 32. After receiving the access address, the serial flash memory 32 will search and return the searched programming codes to the buffering/controlling device 38 and the microprocessor 30(for ~~fasten~~ speeding up the whole access process). The buffering/controlling device 38 then recovers the operating clock for the microprocessor 30 so that the microprocessor 30 can access the programming codes. With the structure and method according to the present embodiment, the microprocessor 30 can execute the codes when accessing the low-speed serial flash memory 32.

[0028] Please refer to Fig.10. In the present embodiment, the program is stored in the random access memory 34(it's usually a dynamic random access memory). We first load the instructions from the serial flash memory 32 to the random access memory 34, then the microprocessor 30 will execute the program via the random access memory 34. The microprocessor 30 does not directly acquire the desired digital data from the random access memory 34 but utilizes the buffering/controlling device 38 beforehand to consecutively access a plurality of digital data in the random access memory 34 instead. The buffering/controlling device 38 outputs an operating clock to the microprocessor 30 so as to slow down/stop microprocessor 30; that is, when the operating clock disappears, the operations of the microprocessor 30 will suspend, when the operating clock [[slow]] slows down, the operations of the microprocessor 30 will also slow down. The buffering/controlling device 38 can consecutively access a predetermined number of programming codes in the random access memory 34. When the microprocessor 30 requires the programming codes and accesses the buffering/controlling device 38, the microprocessor 30 emits an access address corresponding to the programming codes to the buffering/controlling device 38 so that the buffering/controlling device 38 can judge whether the access address emitted from the microprocessor 30 is located in the buffering/controlling device 38.

If the buffering/controlling device 38 stores the programming codes required by the microprocessor 30, the microprocessor 30 directly receives the desired programming codes from the buffering/controlling device 38. On the other hand, if the buffering/controlling device 38 does not store the programming codes required by the microprocessor 30 （for instance, the microprocessor 30 is in jump condition）, the buffering/controlling device 38 will stop the outputting of the operating clock so that the microprocessor 30 will suspend and retain the current conditions due to the disappearance of the operating clock. In the meantime, the buffering/controlling device 38 transmits the access address corresponding to the programming codes to the random access memory 34. After receiving the access address, the random access memory 34 will search and return the searched programming codes to the buffering/controlling device 38 and the microprocessor 30(for ~~fasten~~ speeding up the whole access process). The buffering/controlling device 38 then recovers the operating clock for the microprocessor 30 so that the microprocessor 30 can access the programming codes. With the structure and method according to the present embodiment, the microprocessor 30 can execute the codes when accessing the low-speed random access memory 34.

[0029] Please continue to refer to Fig.9. In the present embodiment, the program is partially stored in the random access memory 34(it's usually a dynamic random access memory) and partially stored in the serial flash memory 32. We first load partial of the instructions from the serial flash memory 32 to the random access memory 34,then the microprocessor 30 will execute the program via the random access memory 34 or the serial flash memory 32.Usually, for faster response speed, we will put frequently used codes into random access memory 34, and if the required instructions are available in the random access memory 34, the buffering/controlling device 38 will fetch the codes from the random access memory 34, else the buffering/controlling device 38 fetch the codes from the serial flash memory 32.In fact, the microprocessor 30 does not directly acquire the desired

digital data from the random access memory 34 or the serial flash memory 32 but utilizes the buffering/controlling device 38 beforehand to consecutively access a plurality of digital data in the random access memory 34 or the serial flash memory 32 instead. The buffering/controlling device 38 outputs an operating clock to the microprocessor 30 so as to slow down/stop microprocessor 30; that is, when the operating clock disappears, the operations of the microprocessor 30 will suspend, when the operating clock slow down, the operations of the microprocessor 30 will also slow down. The buffering/controlling device 38 can consecutively access a predetermined number of programming codes in the random access memory 34 or the serial flash memory 32. When the microprocessor 30 requires the programming codes and accesses the buffering/controlling device 38, the microprocessor 30 emits an access address corresponding to the programming codes to the buffering/controlling device 38 so that the buffering/controlling device 38 can judge whether the access address emitted from the microprocessor 30 is located in the buffering/controlling device 38. If the buffering/controlling device 38 stores the programming codes required by the microprocessor 30, the microprocessor 30 directly receives the desired programming codes from the buffering/controlling device 38. On the other hand, if the buffering/controlling device 38 does not store the programming codes required by the microprocessor 30 （for instance, the microprocessor 30 is in jump condition）, the buffering/controlling device 38 will stop the outputting of the operating clock so that the microprocessor 30 will suspend and retain the current conditions due to the disappearance of the operating clock. In the meantime, the buffering/controlling device 38 transmits the access address corresponding to the programming codes to the random access memory 34 or the serial flash memory 32. After receiving the access address, the random access memory 34 or the serial flash memory 32 will search and return the searched programming codes to the buffering/controlling device 38 and the microprocessor 30(for ~~fasten~~ speeding up the whole access process). The buffering/controlling device 38 then recovers the operating clock for the

microprocessor 30 so that the microprocessor 30 can access the programming codes. With the structure and method according to the present embodiment, the microprocessor 30 can execute the codes when accessing the low-speed random access memory 34 or the serial flash memory 32.

Please replace paragraph [0043] with the following amended paragraph:

[0043] Please continue to refer to Fig.8. The microprocessor emulator 54 is electrically connected to a second memory 53, which can be a program memory such as a static random access memory (SRAM), a flash memory, or a ROM. A plurality of instructions, which are required for operations of the microprocessor emulator 54, are stored in the second memory 53. When the buffering/controlling device 58 outputs the operating clock to the microprocessor emulator 54, the second memory 53 will deliver related instructions to the microprocessor emulator 54. When the buffering/controlling device 58 suspends to output the operating clock to the microprocessor emulator 54, the microprocessor emulator 54 suspends and is unable to receive any instruction from the second memory 53. In addition, the frequency of the operating clock provided by the buffering/controlling device 58 can be adjusted according to the real situations. The frequency of the operating clock can also be adjusted by being electrically connected to an external clock generator 56. In summary, the microprocessor emulator 54 of the present invention can accurately emulate the performances of the microprocessor system 50 combined with the low-speed serial flash memory 52 and a dynamically adjustable operating clock.

Please insert a new paragraphs [0043.1] between original paragraphs [0043] and [0044]:

[0043.1] In the present invention, adjusting an executing speed can be achieved by adjusting an operating clock using an external circuit or a circuit installed in the

microprocessor; achieved by inserting an NOP (No Operation) command among commands; or achieved by keeping a program counter unchanged. In addition, the buffering/controlling device comprises a FIFO storage structure, a dynamic random access memory (DRAM), or a static random access memory (SRAM) for data buffering.